

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Kumpulkan dalam dorm.zip

Pada kali ini anda diminta gro untuk mengimplementasikan sistem informasi untuk pengelolaan asrama di **Nimonsversity**. Gunakan konsep *Object Oriented Programming* yang telah anda pelajari.

Pada **Nimonsversity** terdapat beberapa asrama yang dapat ditinggali. Asrama Internasional untuk mahasiswa asing, Asrama Sangkuriang hanya untuk mahasiswa domestik Asrama Kidang Pananjung hanya untuk mahasiswa domestik pria, dan Asrama Kanayakan hanya untuk mahasiswa domestik wanita.

Ada 2 user dalam sistem informasi ini, Pengelola dan Mahasiswa. Keduanya dapan melakukan login dan register pada sistem. Pada mahasiswa, terdapat opsi login tambahan menggunakan SSO (*simple version*).

Mahasiswa dapat mendaftar ke asrama selama kapasitas asrama masih mencukupi. Setelah itu, mahasiswa yang telah terdaftar dapat diverifikasi oleh pengelola asrama.

Lengkapilah file [berikut](#) untuk membantu gro dalam menyelesaikan sistem informasi **Nimonsversity Dormitory**

**Note:**

- Gunakan String.format untuk melakukan casting angka menjadi string yang diawali beberapa digit 0. Contoh, untuk mengubah 1 menjadi 001 --> String.format("%03d", 1);

- Contoh penggunaan Dorm.zip -> [Driver.zip](#)

- Clue tambahan: gunakan getAsramaInfo pada mahasiswa untuk pengecekan di terima pendaftaran pengelola!

Java 8 ▾

 [dorm.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.07 sec, 28.51 MB
2	10	Accepted	0.06 sec, 28.47 MB
3	10	Accepted	0.07 sec, 28.64 MB
4	10	Accepted	0.06 sec, 29.05 MB
5	10	Accepted	0.06 sec, 28.54 MB
6	10	Accepted	0.07 sec, 28.05 MB
7	10	Accepted	0.07 sec, 26.90 MB
8	10	Accepted	0.06 sec, 28.29 MB
9	10	Accepted	0.07 sec, 28.59 MB

## Question 2

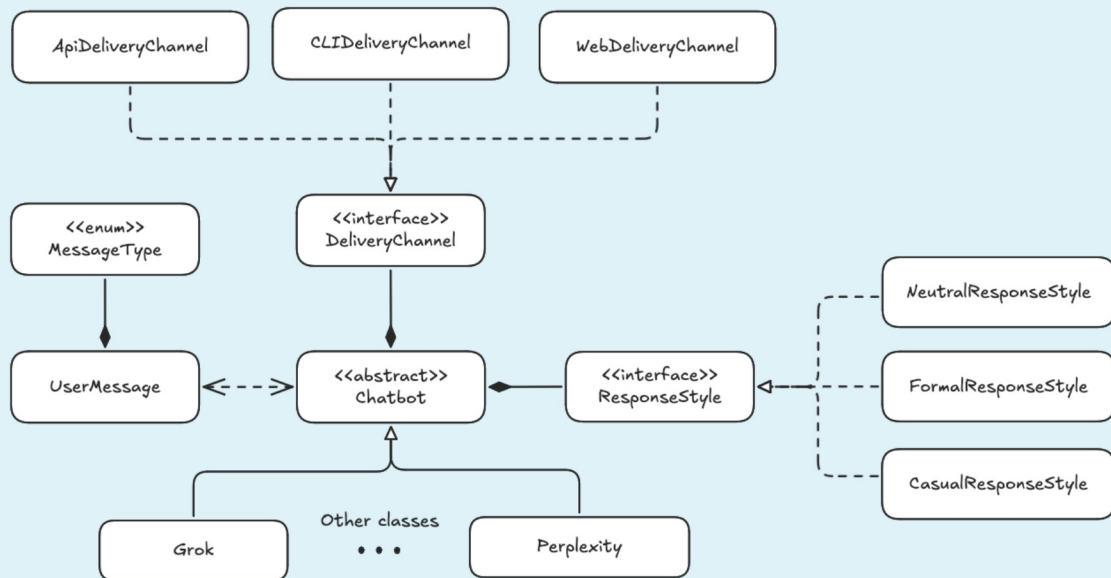
Correct

Mark 100.00 out of 100.00

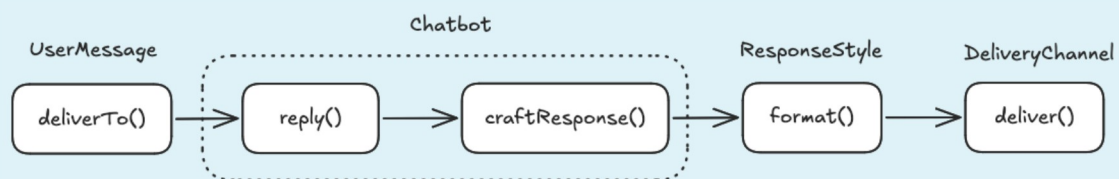
Time limit	1 s
Memory limit	64 MB

**Nama File:** Chatbot.zip

Anda diminta untuk mengembangkan sebuah sistem **ChatBot** dirancang untuk menerima masukan berupa **userMessage** dan memberikan respons yang bervariasi tergantung pada konfigurasi yang digunakan. ChatBot ini dapat **dirakit secara modular** dengan memilih berbagai **ResponseStyle** dan **DeliveryChannel**. Dengan demikian, ChatBot yang sama dapat berperilaku berbeda tanpa mengubah kode utama, cukup dengan mengganti objek-objek implementasi yang dikombinasikan saat pembuatan objek. Berikut adalah gambaran kelas - kelas yang digunakan dalam permasalahan ini.



Urutan flow ketika Chatbot ingin membuat response dapat dilihat pada gambar di bawah ini.



Implementasikan kelas dalam file [Chatbot.zip](#). Upload jawaban sebagai berkas **Chatbot.zip** yang berisi tepat 7 file yaitu: Chatbot.java, CliDeliveryChannel.java, ApiDeliveryChannel.java, WebDeliveryChannel.java, CasualResponseStyle.java, FormalResponseStyle.java, NeutralResponseStyle.java.

**Catatan:** Setiap String yang di-return **tidak disertai newline**. Untuk membantu melakukan pengujian, silahkan gunakan main file yang ada di dalam zip. Di dalam zip tersebut terdapat 2 contoh concrete class dari Chatbot yaitu Grok dan Perplexity. Output yang diharapkan dari kelas main tersebut dapat dilihat di file [berikut](#).

Java 8

[Chatbot.zip](#)

Score: 100

## Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.05 sec, 27.87 MB
2	10	Accepted	0.05 sec, 28.36 MB
3	10	Accepted	0.05 sec, 27.07 MB
4	10	Accepted	0.05 sec, 26.86 MB
5	10	Accepted	0.05 sec, 28.05 MB
6	10	Accepted	0.04 sec, 27.45 MB
7	10	Accepted	0.05 sec, 26.91 MB
8	10	Accepted	0.04 sec, 27.45 MB
9	10	Accepted	0.06 sec, 26.92 MB
10	10	Accepted	0.06 sec, 29.98 MB

[◀ Praktikum 5](#)

Jump to...



[Pra Praktikum 6 ▶](#)